Enhancing AdultSize Humanoid Localization Accuracy: A Vision-based aMCL Leveraging Object Detection Model and Hungarian Algorithm

Jun Young Kim[1], Min Sung Ahn[2] and Jeakweon Han*

***Abstract*— The robot's decision-making is an essential component of the autonomous robot. To fulfill this component, the accurate position estimation of the robot is a fundamental prerequisite. Localization determines the robot's relative position within the map environment, and existing mobile robots have been extensively studied. However, localization is still challenging for humanoids because the bipeds' movement is not as stable as mobile robots, and camera view frames oscillate from side to side. Developing localization with high accuracy under the above-limited constraints is necessary. This paper proposes an estimation of a 1.3m tall humanoid robot's position with an adaptation of vision-based Augmented Monte-Carlo localization (aMCL) in the soccer field. Several approaches were also applied to enhance the localization performance. First, a deep learning-based object detection model was selected to identify pre-defined landmarks. In addition, the data association process was improved from the nearest neighbor matching algorithm to the Hungarian algorithm. This method enhanced data association performance, and the robot's position was successfully estimated in real-time.**

I. Introduction

Knowing own position and orientation is crucial for a soccer player, and it is just as crucial for an autonomous robot plays soccer. If a robot does not know its own position and direction on the playing field, it becomes difficult to distinguish its goal from its opponent's, which could result in unintended own goals in the game.

Humanoid robots must track or estimate their own position. However, this is not as straightforward as it is for humans for several reasons. Firstly, the playing field surface is made of artificial grass, which makes it challenging to maintain a stable walk for the bipedal robot. As a result, simple position tracking techniques such as dead reckoning [1] are unsuitable for humanoid robot systems, as they produce noisy results and are prone to drift over time. Secondly, additional external sensors can be used to estimate the robot's position. However, the humanoids competing in the RoboCup Humanoid Soccer League have to be as human-like as possible, particularly in their kinematic structure and the types of sensors used. For this reason, the RoboCup Federation currently bans using LiDAR and laser-based sensors in humanoid robots, permitting only camera sensors. This restriction makes it more challenging.

[1]Author Jun Young Kim is with the Department of Interdisciplinary Robot Engineering Systems, Hanyang University ERICA, 15588 Ansan, Republic of Korea. lgkimjy@hanyang.ac.kr

[2]Min Sung Ahn is with the Department of Mechanical and Aerospace Engineering at the University of California, Los Angeles, CA 90095, USA. aminsung@ucla.edu

*Jeakweon Han is an associate professor in the Department of Robotics, Hanyang University ERICA, 15588 Ansan, Republic of Korea. jkhan@hanyang.ac.kr

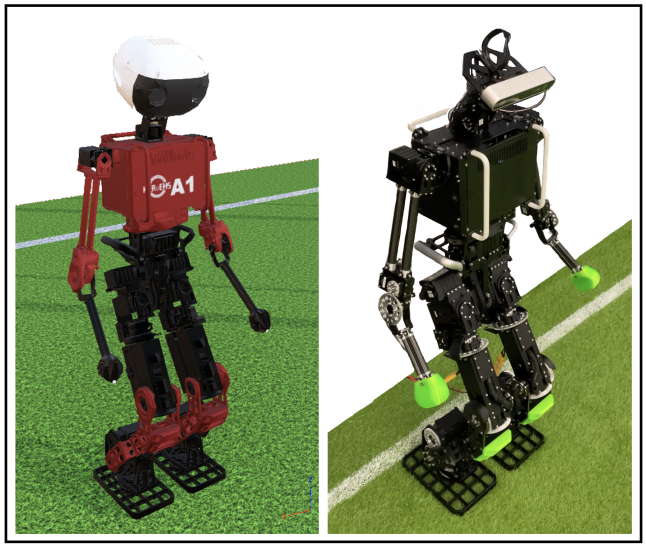

Fig. 1. Both the simulation model (**Left**) and the physical robot (**Right**) of ALICE Version 3. Both models were utilized to conduct the simulation and real-world experiments.

To address these issues, localization methods have been developed in this field. Localization is the process of creating a correlation between the map coordinate system and the robot's local coordinate system. Several approaches exist, including EKF-based Localization and Monte Carlo Localization (MCL). EKF-based localization [2] uses a mathematical model of the system's dynamics and sensor measurements to estimate the robot's state. Another notable method for the humanoid system is MCL [3]–[6], a particle filter and Bayes filter-based algorithm. This method is well-suited for handling noisy input and working with nonlinear distributions. However, this method still has major drawbacks, one being computational complexity. As the number of particles and measured input increases, MCL requires significant computation, making real-time implementation challenging for resource-limited platforms. The other drawback is sensing noise, as MCL relies on accurate sensor data, any measurement noise or errors can lead to degraded performance. Also, it could have trouble converging to the right position if the initial distribution of samples is far from the true position.

The measurement input is an important factor that determines the accuracy of the localization. Generally, in RoboCup Soccer League, approaches involve utilizing field lines and line points as measurement inputs [7]–[11]. However, such approaches need to address computational ef-

ficiency concerns related to the segmentation and feature extraction processes involved in computer vision. These processes can impact real-time localization capabilities and make it challenging to perceive the environment in a timely manner. Moreover, the sampling-based method MCL is sensitive to larger measurement inputs, which can further compromise its real-time performance. To address the limitations of computer vision processing, we have replaced it with a deep learning-based object detection model for measuring inputs. This approach can effectively reduce the size of observation inputs while maintaining robustness in the localization process.

Another important process in localization is data association, which significantly enhances accuracy. The nearest neighbour (NN) algorithm is commonly employed for data association. NN calculates the Euclidean distance and associates it with the data point having the minimum distance. While NN is a fast and intuitive method, it often suffers from mismatching problems because relying solely on the Euclidean distance condition is not always effective for the accurate association. Several approaches have been proposed to address this, such as the Kd-tree, a spatial data structure used for efficient searching of NN. Kd-tree improves the effectiveness of data association by providing a more robust and optimized search mechanism. Another approach, Joint Compatibility Branch and Bound (JCBB) [12], [13] has also been introduced to overcome the limitations of NN-based association. However, JCBB still has some drawbacks, such as computationally demanding compared to NN-based association. Therefore, the Hungarian algorithm [14] was utilized in this paper to achieve optimal association, a well-established optimization method that ensures perfect data association that improves localization accuracy.

This paper proposes an enhanced localization that incorporates an object detection model and the Hungarian algorithm for precise real-time estimation. Additionally, it addresses the challenge of re-estimating the robot's position in the case of the kidnapped problem. In Section 2, the proposed methodology for the humanoid platform will be detailed, explaining the steps and techniques employed in the localization process. The experimental results of the proposed methods for the humanoid robot ALICE3 in both simulation and real-world scenarios will be discussed in Section 3. Section 4 presents the discussion, addressing challenges and limitations and comparing existing approaches. Finally, Section 5 concludes by summarizing the research and emphasizing the contributions of the study.

II. Augmented Monte Carlo Localization

This section presents the Augmented Monte Carlo Localization (aMCL) algorithm for the humanoid system. It is outlined in Algorithm 1, where χ_t represents sets of particle samples, u_t represents control input, and z_t represents measurement input.

Augmented MCL estimates the robot's position through a recursive prediction and correction process. It uses particle samples scattered in the environment, which are updated based on the sample motion model and measured for their likelihood using the measurement model. Samples with low similarity will be discarded during resampling, and high-weighted particles will be reassigned for the next prediction step. Also, aMCL will track the long-term and short-term average weight to survive the kidnap problem.

Algorithm 1 Augmented Monte Carlo Localization

Input: χ_{t-1}, u_t, z_t, m
Output: χ_t

static $\omega_{slow}, \omega_{fast}$
$\bar{\chi}_t = \chi_t = \emptyset$
for $m = 1$ to M **do**
  $x_t^{[m]}$ = sample_motion_model$(u_t, x_{t-1}^{[m]})$
  $\omega_t^{[m]}$ = measurement_model$(z_t, x_t^{[m]}, m)$
  $\chi_t = \bar{\chi}_t + \langle x_t^{[m]}, \omega_t^{[m]} \rangle$
  $\omega_{avg} = \omega_{avg} + \frac{1}{M}\omega_t^{[m]}$
end for
$\omega_{slow} = \omega_{avg} + \alpha_{slow}(\omega_{avg} - \omega_{slow})$
$\omega_{fast} = \omega_{avg} + \alpha_{fast}(\omega_{avg} - \omega_{fast})$
for $i = 1$ to M **do**
  idx = resampling(ω)
  if $max(0.0, 1.0 - \omega_{fast}/\omega_{slow})$ **then**
    add random pose to χ_t
  else
    idx = resampling(ω)
    add $x_t^{[idx]}$ to χ_t
  end if
end for
return χ_t

A. Sample Motion Model

The first step of the aMCL is to obtain x_t of M samples as a prediction step. In probabilistic robotics [3], the odometry and velocity motion model is modelled for mobile robots. However, a new motion model must be developed for the humanoid system. This study used the humanoid robot ALICE3 shown in Fig. 1, 3 to validate the motion model.

The estimation will be performed through dimensionality reduction to three-dimension (x, y, θ). ALICE3 uses a ZMP-based control gait method, which makes the robot's Center of Mass (CoM) follow the reference body trajectory as described in [15], [16]. Equation (1) accumulates the robot's position based on actual kinematic information, where Δ_x, Δ_y, and Δ_θ represent the controlled motion input, the calculated amount of change in the robot's position through kinematics.

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} \cos\theta_p & \sin\theta_p & 0 \\ \sin\theta_p & \cos\theta_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta_{x,noise} \\ \Delta_{y,noise} \\ \Delta_{\theta,noise} \end{bmatrix} + \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} \quad (1)$$

However, in the real world, the backlash of each joint, slippery noise caused by field grass, and other disturbances could make it difficult to reach an ideal position. Therefore, the motion model of M samples' movement will be added with some extra Gaussian noise, Δ_{xnoise}, Δ_{ynoise}, and $\Delta_{\theta noise}$.

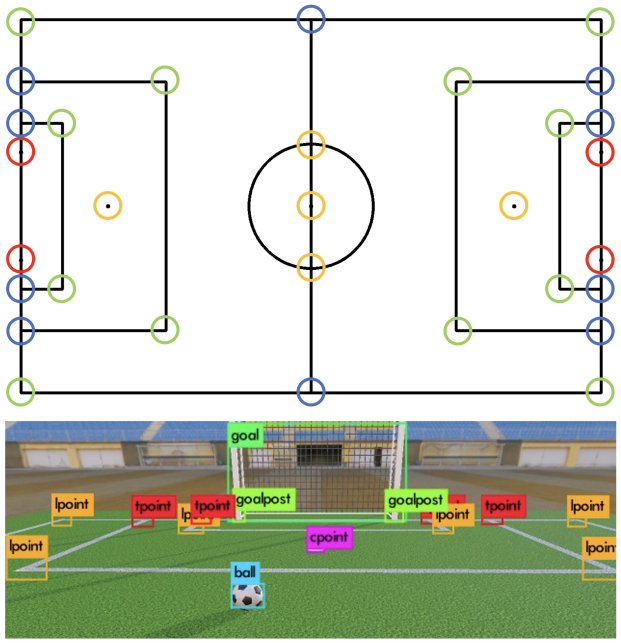

Fig. 2. Top: Outlines pre-defined keypoints on the soccer field, with different shapes indicating different intersection types. Cross-shaped intersections in **orange**, T-shaped intersections in **blue**, L-shaped intersections in **green**, and the goalposts are indicated in **red**. Bottom: Illustrates the keypoints detected via the robot within the Webots simulation.

Gaussian noise to the input control value u_t will be according to the uncertainty that can occur on the x and y-axis of the robot.

B. Measurement Model

After the sample motion model, the measurement model will be processed, designed as outlined in Algorithm 2. It is the step of correcting the prediction of the sample motion model, which measures the weight ω of each particle sample and finds the estimated robot position.

Algorithm 2 Proposed measurement model

Input: $z_t, x_t^{[m]}, m$
Output: ω_t
1: **for** $m = 1$ to M **do**
2: Transformation of observed keypoints
3: Data association (Matching)
4: Multi-variant Gaussian function
5: **end for**
6: **return** ω_t

We used the deep learning-based detection model YOLO to extract four pre-defined keypoints (c-point, l-point, t-point, goalpost) as described in Fig. 2. Labelled over 30,000 datasets, and these extracted keypoints were used for the weight calculation of each particle.

1) Transformation of observed keypoints: Each of the detected keypoints will be transformed using the kinematic information of ALICE3 with a transformation matrix (2). The configuration of the joint axis of ALICE3 can be seen

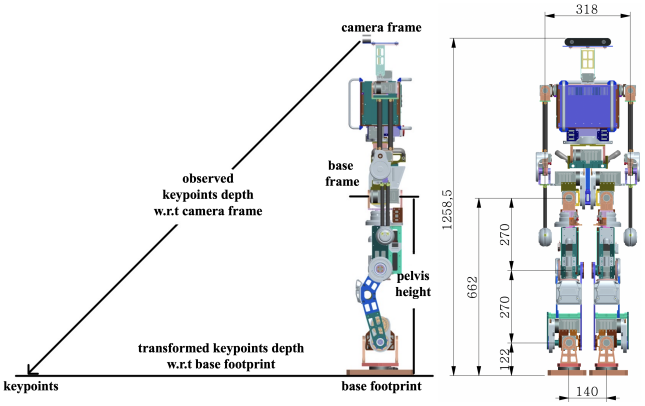

Fig. 3. Left: Visualized transformation between the camera frame and the base footprint. Shows how detected keypoints w.r.t. the camera frame can be converted to points relative to the base footprint. Right: The overall hardware structure of the humanoid robot platform, ALICE ver. 3.

in Fig. 3. For stable walking, ALICE3 maintains a fixed pelvis height. Therefore, only the homogeneous transform matrix from the pelvis joint to the camera frame needs to be considered as (3), where ψ represents yaw, and θ represents pitch. By multiplying each detected keypoints with the transform matrix T_{pelvis}^{camera}, the distance of the keypoints will be calculated from the base footprint of the robot, not the camera.

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$T_{pelvis}^{camera} = T_{head\theta}^{camera}\ T_{head\psi}^{head\theta}\ T_{torso\psi}^{head\psi}\ T_{torso\theta}^{torso\psi}\ T_{pelvis}^{torso\theta} \tag{3}$$

Finally, the distance from the base footprint of the robot to the keypoints is transformed to the keypoint of each particle sample based on the global coordinate system using (4).

$$\begin{bmatrix} x'_{obs} \\ y'_{obs} \end{bmatrix} = \begin{bmatrix} \cos(\theta_p) & -\sin(\theta_p) \\ \sin(\theta_p) & \cos(\theta_p) \end{bmatrix} \begin{bmatrix} x_{obs} \\ y_{obs} \end{bmatrix} + \begin{bmatrix} x_p \\ y_p \end{bmatrix} \tag{4}$$

2) Data Association: The data association process is crucial as it matches the keypoints with the pre-defined landmarks to calculate weights (likelihoods). These associations are essential for accurately measuring the likelihood by correctly matching landmarks with keypoints. As introduced earlier, we employed the nearest neighbor (NN) algorithm to associate the landmarks and observed keypoints. This method calculates the Euclidean distance between each observed keypoint and all 31 landmarks, assigning each keypoint to the landmark with the minimum distance. However, as shown in Fig. 4, this approach can result in mismatching problems. As data association is a critical step for correction, these mismatches could adversely affect localization performance.

To overcome this issue, we used the Hungarian algorithm, an efficient and optimal full-matching method for assignment problems [14], [17], for optimal matching. This algorithm finds the optimal solution by employing equations (6) and (7). The Hungarian algorithm significantly mitigates the

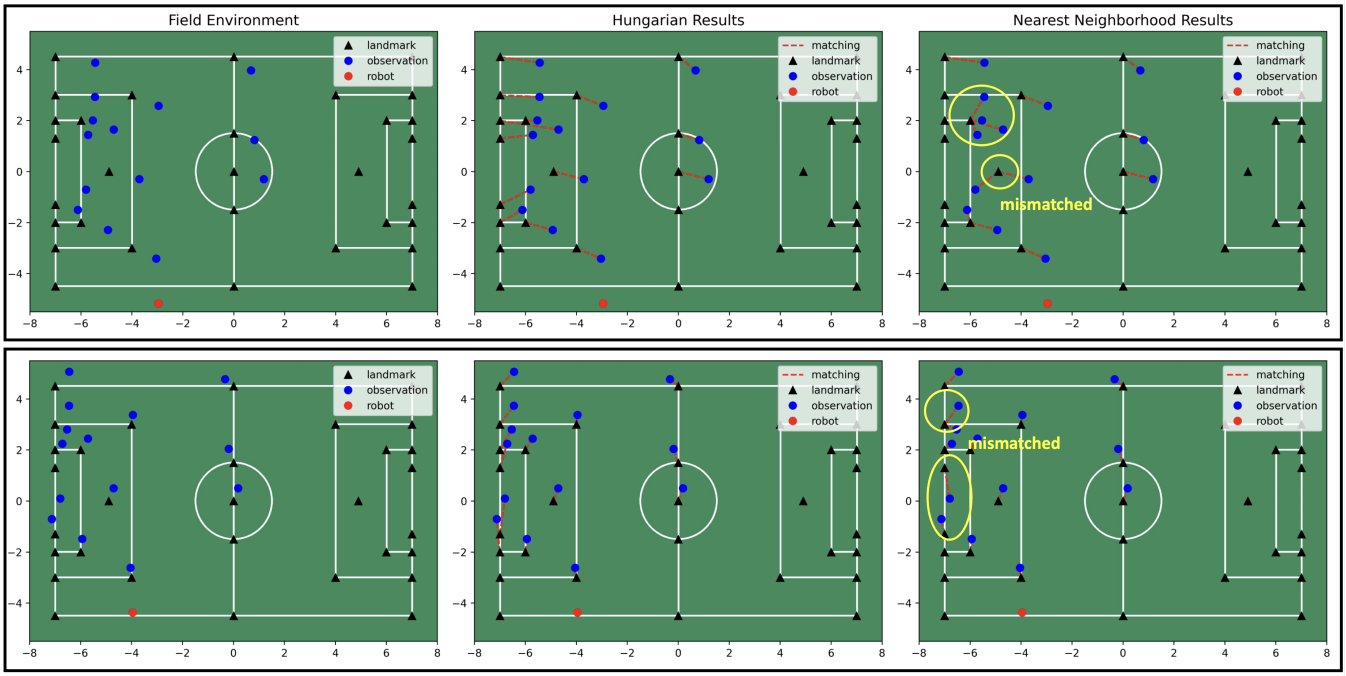

Fig. 4. The results of data association when using the NN and Hungarian methods, and each row represents each scenario. Left: the transformed keypoints (**blue**) as observed based on the position of the sample (**red**). Middle: Presents the results of matching keypoints to landmarks using the Hungarian algorithm. Right: the results of matching keypoints to landmarks using the NN. As evidenced by the matching results, while the Hungarian method results in perfect matches, the NN method shows several mismatched results.

mismatching problem, enhancing the reliability and accuracy of the localization process.

$$\text{Cost} = \begin{matrix} c_{1,1} & c_{1,2} & \dots & c_{1,j} \\ c_{2,1} & c_{2,2} & & c_{2,j} \\ \vdots & \vdots & & \vdots \\ c_{i,1} & c_{i,2} & \dots & c_{i,j} \end{matrix} \tag{6}$$

$$\text{minimize} \quad \sum_{i,j} \text{Cost}(i,j) \tag{7}$$

As shown in Fig. 4, the Hungarian algorithm clearly outperforms the NN algorithm, providing more precise and reliable matching results, thereby enhancing the accuracy of the localization process.

3) Multi-variant Gaussian: The last step of the measurement model is to calculate the samples' weight using the assigned data obtained through the data association process. The measurement data z_t is expressed by (8).

$$z_t = \{(x,y)_t^1,\ (x,y)_t^2, \dots\ , (x,y)_t^k\} \tag{8}$$

A multivariate Gaussian function (9) is applied to measure the weight of a corresponding sample, where x'_{obs} and y'_{obs} denote the positions of the transformed observed keypoints, while μ_x and μ_y represent the positions of the corresponding landmarks on the soccer field.

$$\omega(x) = (2\pi\sigma_x\sigma_y)^{-\frac{1}{2}} \exp\{-(\frac{(x'_{obs}-\mu_x)^2}{2\sigma_x^2} + \frac{(y'_{obs}-\mu_y)^2}{2\sigma_y^2})\} \tag{9}$$

Each of σ_x, σ_y in (9) are parameters that determine the uncertainty of the measurement sensor. These values are determined according to camera noise and depth accuracy. We must experimentally determine how much the distance error occurs when the camera depth noise is the most severe.

$$\omega(x) = \prod_{k=1}^{K} \omega(x) \tag{10}$$

Every keypoints that the sample observed has to be calculated by (9), and the final weight that the sample will be determined by (10), where K is the number of observed keypoints. The highest weighted particle will be the estimated position of the robot.

C. Resampling

After the correction process, a resampling step is performed according to each current sample's weight to predict the following motion, which means that the new M samples will be extracted from the previous sample set through resampling.

One of the genetic algorithms called the roulette wheel algorithm was selected for the resampling, as outlined in Algorithm 3. The roulette wheel algorithm has a time complexity of $M \times O(logM)$, which has a higher time complexity than the low variance sampler and stochastic universal sampling methods [18]. However, it is still appropriate to implement due to the size of the sample sets was not large enough to be adversely affected. The roulette wheel algorithm could resample the sample sets with high weights efficiently and intuitively. The survived sample sets will be returned to be used for the next prediction.

Algorithm 3 Roulette Wheel Resampling

Input: Set of weights for each particle ω
Output: Set of resampled weights for each particle $\omega_{sampled}$

1: $\omega_{sampled} = \emptyset$
2: idx = integer number of $rand() * M$
3: $\beta = 0.0$
4: $\omega_{max} = max(\omega)$
5: **for** $i \leftarrow 0$ To M **do**
6: $\quad \beta = \beta + rand() * 2.0 * \omega_{max}$
7: $\quad$ **while** $\beta > \omega^{idx}$ **do**
8: $\quad\quad \beta = \beta - \omega^{idx}$
9: $\quad\quad idx = (idx + 1) \% M$
10: $\quad$ **end while**
11: $\quad$ add ω^{idx} to $\omega_{sampled}$
12: **end for**
13: **return** $\omega_{sampled}$

D. Weight Tracking

Due to the texture of the grass on a soccer field, a humanoid robot could easily lose its position. Also, unexpected position errors can occur in a struggle for ball competition with an opponent robot. For these reasons, tracking and estimating humanoid position is much more challenging than mobile robots. In such environment, the estimation might fail or the robot may be kidnapped because of unexpected physical contact. In other words, samples might not survive after the resampling process. To solve certain situations, aMCL measures likelihood by averaging the weights of the existing sample set. Lines 9 and 10 of the Algorithm 1 maintain the long- and short-term average weights. The parameters $\alpha_{slow}, \alpha_{fast}$ determine the decay rates for the exponential filter that calculates the long- and short-term weight averages.

$$max(0.0,\ 1.0 - \omega_{fast}/\omega_{slow}) \tag{11}$$

Obtained long- and short-term weights will be used to calculate decay rates. After that, samples scatter at random locations according to the calculated probability using (11).

III. Experiments

This section outlines the experimental setup and assesses the results and performance of the applied algorithms through both simulated and real-world tests. This experimental evaluation primarily focuses on two aspects: First, it assesses the ability to estimate the robot's position in real-time using object detection models instead of using the lines and points derived from the computer vision process. Secondly, it emphasizes evaluating how much the localization performance improves when the Hungarian algorithm is used instead of standard nearest-neighbor methodologies.

A. Simulation Setup

The verification and performance evaluation of the algorithm was initially conducted through simulation using Webots as the simulation tool, with the environment designed to follow the actual RoboCup 2022 rules. The algorithm was entirely written in Python, and the simulation environment was operated on Ubuntu 20.04.

B. Simulation

We designed four scenarios of walking on the soccer field, where we tested both the NN and the Hungarian algorithm. Table I represents the success rate of matching for each method. As can be seen from the table, the Hungarian algorithm demonstrates superior performance in data association. On the other hand, the NN shows poor performance.

TABLE I
SUCCESS MATCHING RATE OF EACH METHOD PER SCENARIO

index	Success Rate (nearest)	Success Rate (hungarian)
scene 1	80.89%	**90.00%**
scene 2	84.25%	**92.01%**
scene 3	83.98%	**93.22%**
scene 4	85.90%	**96.85%**

Fig. 5 is the results of the robot's position estimation using Augmented MCL for both **Scenario 2** and **Scenario 4**. The notation in Fig. 5 is as follows: Red represents the robot's position, which is accumulated from the kinematic data. Blue denotes the ground truth value, measured with a GPS sensor mounted on the robot's pelvis, and yellow indicates the estimated robot position using aMCL, implemented with the Hungarian algorithm. The inclusion of kinematically accumulated data in the figure is to highlight the inaccuracies in humanoid robot motion on a soccer field. As can see from the results, it can be confirmed that the ground truth value and the starting point of kinematics are the same. However, after a few walks on the soccer field, the kinematically accumulated position experiences a significant error and reaches a completely different position. On the other hand, it can be confirmed that the applied aMCL algorithm estimates the robot's position accurately.

TABLE II
RMSE AND MIN/MAX ERRORS FOR EACH METHOD AND SCENARIO

index	RMSE (m)		Mininum Error (m)		Maximum Error (m)	
	nearst	optimal	nearst	optimal	nearst	optimal
scene 1	0.275	**0.234**	0.008	**0.006**	0.529	**0.607**
scene 2	0.293	**0.169**	0.030	**0.013**	0.698	**0.309**
scene 3	0.348	**0.237**	0.011	**0.015**	0.987	**0.809**
scene 4	0.290	**0.208**	0.014	**0.023**	0.888	**0.673**

Table 2 effectively underscores the notable improvement in the performance of our localization method. It presents a comprehensive analysis of the algorithm's ability to accurately estimate the robot's position, which is a critical aspect of autonomous navigation. The RMSE recorded in these position estimations is best at 0.169m. This level of accuracy in such a dynamic and complex environment illustrates the robustness and effectiveness of our applied methods.

In addition, we evaluated our algorithms under a kidnap scenario to assess their performance under unexpected circumstances. In this unique test setup, the robot is 'kidnapped'

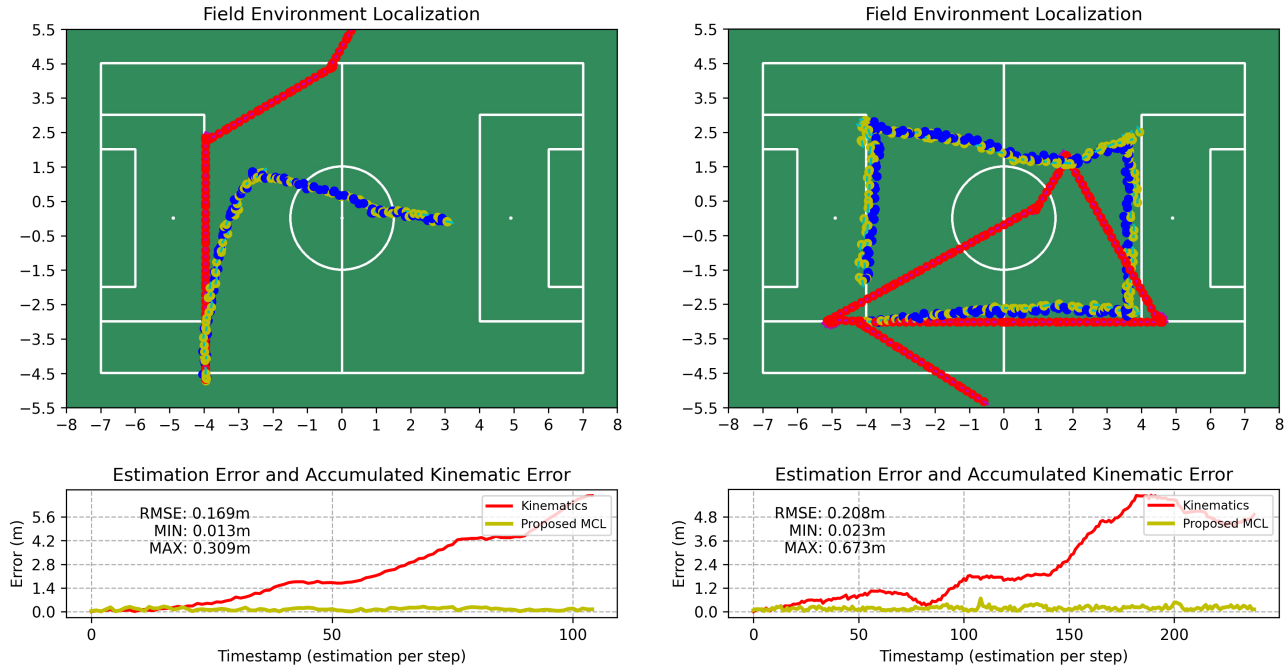

Fig. 5. The localization results from two tested scenarios (scenarios 2, 4) in a simulated environment. The soccer field units in meters. Left: Scenario 2's results are displayed. The estimated position of the robot is shown in **yellow** on the field image, which follows the **blue** (ground truth) well, in contrast to the **red**, which represents the kinematically accumulated position. The figure at the bottom shows the error across all steps and the RMSE of estimation. Right: Scenario 4's results, where the estimated position of the robot closely follows the ground truth.

at the 58th step, as depicted in Fig. 6. This scenario is designed to challenge the algorithm's ability to recover and accurately re-localize the robot after sudden, unaccounted changes in its position. The robot demonstrated a slower recovery to its original position when using NN under given conditions. It was found that the algorithm was not as efficient in quickly re-estimating the robot's position after the 'kidnap,' resulting in a more prolonged period of inaccurate localization. In contrast, the Hungarian algorithm exhibited superior performance in this respect. Its matching methodology allowed for a quicker recovery and more accurate re-localization after the robot's sudden displacement.

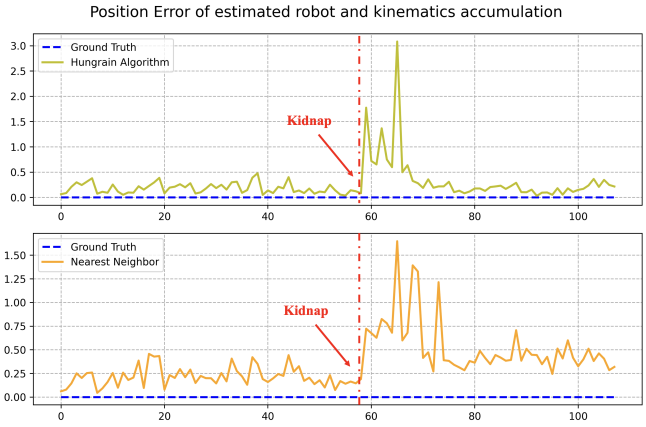

Fig. 6. A simulated scenario of self-localization after the kidnap occurs at step 60. Top: Depicts when the Hungarian algorithm is used, and shows faster self-localization with precision. Bottom: The result with the NN shows a slower recovery and greater positioning errors.

C. Real-World Experiment Setup

The developed aMCL has proven to work effectively in a simulated environment using Webots. Moreover, localization based on detecting keypoint using YOLOv4-tiny was also tested in a real-world setup. We used a compact soccer field for the real-world localization performance test adhering to the official dimensions stipulated for the RoboCup 2018. The detailed configuration of ALICE3, which was tested in this real-world environment, is as follows: Xavier and NUC served as the computational resources, with Ubuntu 20.04 as the operating system. The Robot Operating System (ROS) middleware facilitated data feedback across the control and sensor systems.

D. Real-World Experiment

Fig. 7 depicts the perspective of the humanoid robot during the localization process. The keypoints observed in this process form the crucial input data for the localization.

Fig. 8 shows the result of the estimated robot's position using aMCL in the real-world. The notation within the figure is as follows: Orange represents the robot's position, which is accumulated from the kinematic data. Red is the robot's position accumulating the center point between the legs of the robot, black is the ground truth value of the robot measured by tape measure per 10 cm of movement, and blue is the estimated robot position. Each scenario was chosen to assess the humanoid's possible walking motions on the soccer field

Fig. 7. Top: Demonstrates the testing of localization performance using an actual robot. Bottom: The results of keypoint detection using the YOLOv4-tiny model in the real-world. Highlight the keypoints successfully identified and extracted.

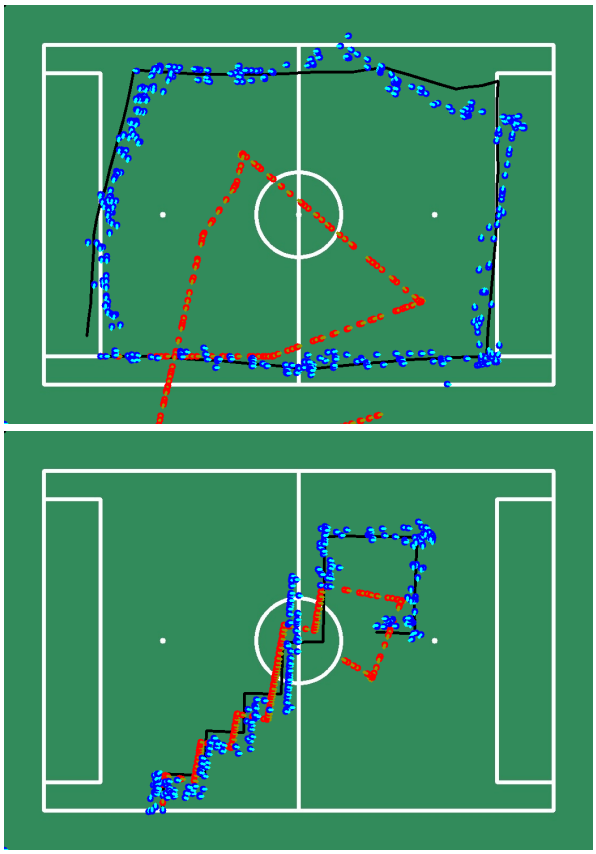

Fig. 8. Two tested scenarios in a real-world environment. Top: Shows the localization results when the robot has completed an entire circuit of the soccer field. Bottom: Depicts the localization results when the robot has moved in a zigzag pattern.

and verify if the robot can navigate the entire field. We can confirm that the ground truth value and the starting point are the same. However, after a few walks on the field, the kinematically accumulated position has a significant error and has reached a completely different position. Unlikely, applied aMCL estimates the robot's position accurately.

It can also be observed that some of the position is misestimated in the middle of the walking on the soccer field. It is likely due to insufficient keypoints detected, or the accuracy of localization is reduced due to errors caused by the oscillation of the view frame received from the camera as the robot moves. Nevertheless, once sufficient measurement data is obtained, the robot's position can re-localized.

IV. Discussion

This section addresses limitations encountered during the experiments. Additionally, it presents comparisons with existing approaches or benchmarks in the field, providing insights into the strengths and weaknesses of the proposed methodology.

The Hungarian algorithm is a one-to-one matching algorithm with significant merit but can also be a potential drawback. The moment one or two mismatched, the rest of the keypoints can also lead to a large number of matching failures in a chain. It could be critical in a soccer field in which all four quarters of the soccer field are symmetrical so that incorrect matching can misestimate the robot's position. Conversely, the NN may be less accurate compared to the Hungarian algorithm. However, it does not suffer from matching failures in a chain. Therefore, the Hungarian and NN should be selected to consider the result of camera calibration, depth accuracy, and size of the soccer field.

Comparing our method to other approaches in the soccer field context, as shown in Table 3, Schulz et al. [8] demonstrated a median average error of 26cm for field crossing and a lower error of 17cm for field lines. Furthermore, Laue et al. [7] achieved an average localization error below 15cm by employing reliable sensor models based on field lines. In another study by Sheikh et al. [19], the application of ORB-SLAM to humanoid robots resulted in an Absolute Trajectory Error (ATE) of 0.19m and 0.23m.

TABLE III
Comparison of Accuracy with Other Methods

methods	[8] (w/ line)	[8] (w/o line)	[19] scene 1	[19] scene2	**ours**
error	0.26m	0.17m	0.19m	0.23m	**0.16m**

However, these approaches have primarily focused on KidSize humanoids, which may have lower levels of camera oscillation and motion errors compared to AdultSize humanoids. Comparing our approach with such methods, we can observe that our performance is considerable. This

comparison provides valuable insight into the performance of our method relative to existing approaches and highlights the potential for improved accuracy in localization.

Despite these advancements, localization methods continue to present numerous challenges. Currently, the robot's initial position is specified by RoboCup, making global localization [20] unnecessary. However, the RoboCup Federation's goal is to facilitate a soccer match with a human by 2050 [21]. By this time, the development of a global localization method, capable of estimating the robot's position without prior knowledge of its initial location, will become indispensable. This task is particularly challenging, given that only a limited number of landmarks can be extracted from the soccer field and that the field is symmetrical in top-bottom and left-right directions.

Furthermore, in real RoboCup competitions, there's a potential issue of opposite robots obscuring keypoints. Future considerations will need to address how to handle keypoints that are blocked by the robots themselves.

Additionally, as the size of the RoboCup soccer field expands each year, the increase in the number of particles, corresponding to the enlarged environment to be estimated, becomes inevitable. To address this, the future will necessitate an algorithm capable of dynamically adjusting the number of particles, such as Adaptive Monte Carlo localization based on kld-sampling.

V. Conclusion

This paper applied the Augmented Monte Carlo localization using the deep learning-based object detection model and optimal assignment algorithm to the humanoid robot, allowing it to accurately track and estimate its real-time position on the soccer field. Additionally, using the Hungarian algorithm significantly enhanced the performance of localization accuracy. In conclusion, good and robust localization is the key to success in Robocup. The developed method contributed to winning **2nd place** at the RoboCup Bangkok 2022 and Bordeaux 2023 AdultSize humanoid robot soccer league. Moving towards future competitions, such as RoboCup 2050, our research is to further refine our methods, aiming for increased accuracy and robustness in the ever-evolving field of humanoid robotics.

Acknowledgment

This research was supported by the MOTIE (Ministry of Trade, Industry, Energy) in Korea, under the Fostering Global Talents for Innovative Growth Program (P0017311), supervised by the Korea Institute for Advancement of Technology (KIAT).

References

[1] B.-S. Cho, W.-s. Moon, W.-J. Seo, and K.-R. Baek, "A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding," *Journal of mechanical science and technology*, vol. 25, no. 11, pp. 2907–2917, 2011.

[2] G. Oriolo, A. Paolillo, L. Rosa, and M. Vendittelli, "Vision-based odometric localization for humanoids using a kinematic ekf," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 153–158.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, ser. Intelligent robotics and autonomous agents. MIT Press, 2005.

[4] W. Hong, C. Zhou, and Y. Tian, "Robust monte carlo localization for humanoid soccer robot," in *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2009, pp. 934–939.

[5] A. Hornung, K. M. Wurm, and M. Bennewitz, "Humanoid robot localization in complex indoor environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1690–1695.

[6] A. C. Almeida, A. H. Costa, and R. A. Bianchi, "Vision-based monte-carlo localization for humanoid soccer robots," in *2017 Latin American robotics symposium (LARS) and 2017 Brazilian symposium on robotics (SBR)*. IEEE, 2017, pp. 1–6.

[7] T. Laue, T. J. De Haas, A. Burchardt, C. Graf, T. Röfer, A. Härtl, and A. Rieskamp, "Efficient and reliable sensor models for humanoid soccer robot self-localization," in *Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the*. Citeseer, 2009, pp. 22–29.

[8] H. Schulz and S. Behnke, "Utilizing the structure of field lines for efficient soccer robot localization," *Advanced Robotics*, vol. 26, no. 14, pp. 1603–1621, 2012.

[9] I. Nagi, W. Adiprawita, and K. Mutijarsa, "Vision-based monte carlo localization for robocup humanoid kid-size league," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, 2014, pp. 1433–1438.

[10] A. Muzio, L. Aguiar, M. R. Máximo, and S. C. Pinto, "Monte carlo localization with field lines observations for simulated humanoid robotic soccer," in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*. IEEE, 2016, pp. 334–339.

[11] G. Ficht, D. Pavlichenko, P. Allgeuer, H. Farazi, D. Rodriguez, A. Brandenburger, J. Kürsch, M. Schreiber, and S. Behnke, "Grown-up nimbro robots winning robocup 2017 humanoid adultsize soccer competitions," in *RoboCup 2017: Robot World Cup XXI 11*. Springer, 2018, pp. 448–460.

[12] X. Shen, E. Frazzoli, D. Rus, and M. H. Ang, "Fast joint compatibility branch and bound for feature cloud matching," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1757–1764.

[13] Y. Luo, C. Yang, and Y. Zhang, "Slam data association of mobile robot based on improved jcbb algorithm," in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 2020, pp. 363–368.

[14] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[15] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1620–1626.

[16] C. Jing and J. Zheng, "Stable walking of biped robot based on center of mass trajectory control," *Open Physics*, vol. 18, no. 1, pp. 328–337, 2020.

[17] E. Reinowski, B. Schultz, J. Wiemers, *et al.*, "Evaluation of further training programmes with an optimal matching algorithm," *Swiss Journal of Economics and Statistics*, vol. 141, no. 4, pp. 585–616, 2005.

[18] K. Jebari and M. Madiafi, "Selection methods for genetic algorithms," *International Journal of Emerging Sciences*, vol. 3, no. 4, pp. 333–344, 2013.

[19] R. Sheikh, S. OBwald, and M. Bennewitz, "A combined rgb and depth descriptor for slam with humanoids," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1718–1724.

[20] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 588–594.

[21] H.-D. Burkhard, D. Duhaut, M. Fujita, P. Lima, R. Murphy, and R. Rojas, "The road to robocup 2050," *IEEE Robotics & Automation Magazine*, vol. 9, no. 2, pp. 31–38, 2002.